

УДК 681.513.675

М.С. Качановська, І.В. Жданова

РОЗПАРАЛЕЛЮВАННЯ ОБЧИСЛЕНЬ У ЗАДАЧІ КЕРУВАННЯ ОДНОСТОРОННІМИ ПРОЦЕСАМИ ДИФУЗІЇ ТА ТЕПЛОМАСО-ОБМІНУ

Вступ

Значну роль при моделюванні та керуванні промислово важливими процесами дифузії та тепломасообміну відіграє ефективність обчислень. Навіть у сучасних умовах при використанні потужних ЕОМ не завжди можна досягти прийнятної часу розв'язання задачі. Це стосується обчислень, пов'язаних із розрахунками на сітках великої розмірності.

Для опису односторонніх процесів дифузії та тепломасообміну, таких, як фільтрація нафти при наявності граничного градієнта тиску, поширення шкідливих домішок при обмеженні на пороговий стан концентрації, як математичні моделі ефективно використовують варіаційні нерівності [1, 2]. Розв'язанню задачі керування такими процесами притаманні обчислювальні складнощі, які пов'язані з великою розмірністю систем рівнянь, що виникають у результаті дискретизації, необхідністю багаторазового розв'язання оптимізаційної задачі та ін.

Питання підвищення обчислювальної ефективності моделювання та керування односторонніми процесами досліджувалися в [3, 4], однак не було розглянуто можливості застосування розпаралелювання обчислень.

Постановка задачі

Мета даної статті полягає в тому, щоб побудувати алгоритм паралельного керування односторонніми процесами з перешкодою в області та експериментально визначити вигоду у часі, що витрачається при паралельній реалізації на кластері ЕОМ порівняно із звичайним "однопотокним" алгоритмом.

Вихідні дані

Як процес, для якого будемо будувати паралельний алгоритм керування, розглянемо фільтрацію нафти при наявності граничного градієнта тиску в просторовій області Ω з границею Σ у проміжок часу $[0, T]$:

$$\left(n(z) \frac{\partial u}{\partial t}, v - u \right) + (Au, v - u) + \psi(v) - \psi(u) \geq \geq (f, v - u) \quad \forall v \in H^1(\Omega) \quad (1)$$

з початковими умовами

$$u(0, z) = u_0(z), \quad z \in \Omega,$$

де оператор заданий таким чином:

$$A(\cdot) = - \sum_{i=1}^3 \frac{\partial}{\partial z_i} \left(b(z) k(z) \frac{\partial}{\partial z_i} \right),$$

і

$$n(z) = \beta_c(z) + m(z)\beta_n(z),$$

де $\beta_c(z)$ — коефіцієнт стисливості пористого середовища; $\beta_n(z)$ — коефіцієнт стисливості нафти; $m(z)$ — щільність нафти; $b(z) = \frac{\rho}{\mu}$, ρ — густина нафти, μ — в'язкість нафти; k — коефіцієнт проникності середовища.

Функція $f(t, z) = \sum_{j=1}^K q_j(t) \delta(z - z^j)$ — примусова функція процесу; $q_j(t)$ — дебіти продуктивних свердловин, що діють у підобластях $\Omega_j \subset \Omega$, $j = 1, \dots, K$, K — кількість свердловин, функціонал відповідності варіаційної нерівності визначається виразом

$$\psi = \begin{cases} \frac{1}{2} \xi(t, z) u^2, & \left| \frac{\partial u}{\partial z} \right| \leq u_{\text{гран}}, \\ 0, & \left| \frac{\partial u}{\partial z} \right| > u_{\text{гран}}. \end{cases}$$

Нехай функціонал, що визначає мету керування процесом (стан u_c), має такий вигляд:

$$J_1 = \frac{1}{2} \int_{\Omega} (u(T, z) - u_c(z))^2 dz \rightarrow \inf_f. \quad (2)$$

Задача полягає в пошуку примусової функції $f \in F$, що реалізувала б мету (2) для моделі (1).

Розв'язання задачі

Для побудови "однопотокного" алгоритму будемо використовувати метод функціональної параметризації, запропонований в [1].

Для цього запишемо нерівність (1) у точковій формі (будемо вважати, що $n(z) = 1$):

$$\begin{aligned} \frac{\partial u}{\partial t} + Au + \xi u &= f, \\ u(z, 0) &= u_0(z), \quad z \in \Omega, \\ u(z, t) &= 0, \quad z \in \Sigma, \end{aligned} \quad (3)$$

де ξ належить простору параметрів Ξ та знаходиться з умови для критерію відповідності:

$$J_0 = \int_0^T \int_{\Omega} \left\{ \begin{aligned} &\left(\frac{\partial u}{\partial t} \right)^2, \left| \frac{\partial u(t, z)}{\partial z} \right| \leq u_{\text{lim}} \\ &\left(\xi \frac{\partial u}{\partial t} \right)^2, \left| \frac{\partial u(t, z)}{\partial z} \right| > u_{\text{lim}} \end{aligned} \right\} dz dt \rightarrow \inf. \quad (4)$$

Задача (3), (4) розв'язується методом Лагранжа. При необхідності керування метод модифікується таким чином: задача керування розв'язується для кожного фіксованого значення параметра ξ , що модифікується згідно з алгоритмом градієнтного спуску.

Запишемо основні співвідношення задачі. Лагранжіан задачі моделювання має вигляд

$$L = J_0 + \left(\frac{\partial u}{\partial t} + Au + \xi u - f, p \right). \quad (5)$$

Тоді умови мінімуму (5) запишуться як

$$\left(\frac{\partial L}{\partial u}, \delta u \right) = 0, \quad (6)$$

$$\left(\frac{\partial L}{\partial \xi}, \delta \xi \right) = 0. \quad (7)$$

Задачу мінімізації (5) розв'язуватимемо методом проєкції градієнта:

$$\xi_{i+1} = \text{Pr}_{\Xi} \left(\xi_i - \lambda \left(\frac{\partial L}{\partial \xi} \right)_i \right). \quad (8)$$

З умови (7) одержимо

$$\frac{\partial L}{\partial \xi} = 2\xi \begin{cases} 0, & \left| \frac{\partial u(t, z)}{\partial z} \right| \leq u_{\text{lim}} + up, \\ \left(\frac{\partial u}{\partial t} \right)^2, & \left| \frac{\partial u(t, z)}{\partial z} \right| > u_{\text{lim}} + up, \end{cases}$$

де p знаходиться з умови (6) як розв'язок спряженої задачі

$$\begin{aligned} -\frac{\partial p}{\partial t} + Ap + \xi p &= 2 \frac{\partial^2 u}{\partial t^2} \begin{cases} 1, & \left| \frac{\partial u(t, z)}{\partial z} \right| \leq u_{\text{lim}}, \\ \xi^2, & \left| \frac{\partial u(t, z)}{\partial z} \right| > u_{\text{lim}}, \end{cases} \\ p|_{t=T} &= 0, \\ p|_{\Sigma} &= 0. \end{aligned} \quad (9)$$

Розв'язок задачі керування отримується аналогічно:

$$\begin{aligned} L^* &= J_1 + \left(\frac{\partial u}{\partial t} + Au + \xi u - f, p \right), \\ f_{i+1} &= \text{Pr}_F \left(f_i - \lambda \left(\frac{\partial L}{\partial f} \right)_i \right), \\ \frac{\partial L}{\partial f} &= -s, \end{aligned} \quad (10)$$

де s знаходиться із співвідношень

$$-\frac{\partial s}{\partial t} + As + \xi s = 0, \quad (11)$$

$$\begin{aligned} s(z, t)|_{t=T} &= u_c(z) - u(z, T), \\ s|_{\Sigma} &= 0. \end{aligned}$$

Таким чином, алгоритм розв'язання задачі передбачає розв'язання двох “вкладених” задач методом градієнтного спуску.

Чисельна апроксимація

Після дискретизації співвідношень (3), (9), (11) згідно з методом скінченних різниць (неявна схема Кранка–Ніколсона) приходимо до необхідності розв'язання великих систем лінійно алгебричних рівнянь (СЛАР). Паралельну організацію деяких методів, що використовуються при розв'язанні СЛАР великого порядку, зокрема методу Гаусса, методів, що базуються на декомпозиції вихідної матриці, розглянуто в [4, 6]. Але існує можливість такої побудови різницевої схеми, де застосовується більш зручний для розпаралелювання метод, наприклад метод прогонки.

Виходячи з цих міркувань, для числової реалізації диференціальних рівнянь у частинних похідних параболічного типу використаємо один з економічних методів реалізації сіткових рівнянь – метод сумарної апроксимації з цілим

кроком [7, 9], який дає змогу одержати матрицю з щільною стрічковою структурою та ефективно застосувати метод прогонки.

Метод сумарної апроксимації для задачі вигляду

$$\frac{du}{dt} + Lu = \varphi, \quad (12)$$

$$u(0) = u_0,$$

де оператор L можна подати як суму операторів $L = \sum_{\alpha=1}^m L_{\alpha}$, полягає в тому, що на кожному кроці часу $[t_n, t_{n+1}]$ розв'язується така послідовність рівнянь:

$$\frac{dv^{\alpha}}{dt} + L_{\alpha}u = \varphi_{\alpha}, \quad \alpha = \overline{1, m}, \quad (13)$$

$$v^1(t_n) = u(t_n),$$

$$v^{\alpha}(t_n) = v^{\alpha-1}(t_{n+1}), \quad \alpha = \overline{2, m},$$

$$u(t_{n+1}) = v^m(t_{n+1}),$$

$$\sum_{\alpha=1}^m \varphi_{\alpha} = \varphi.$$

У загальному випадку схема апроксимує задачу (12) з першим порядком за часом та другим за просторовою координатою. Проте при використанні симетризованого розщеплення оператора L [9]

$$\begin{aligned} \frac{1}{2}L_1 \rightarrow \frac{1}{2}L_2 \rightarrow \dots \rightarrow \frac{1}{2}L_{\alpha} \rightarrow \frac{1}{2}L_{\alpha} \rightarrow \\ \rightarrow \frac{1}{2}L_{\alpha-1} \rightarrow \dots \rightarrow \frac{1}{2}L_1 \end{aligned}$$

чисельні експерименти демонструють кращу точність. Зокрема, для задач (3), (9), (11) оператор має вигляд

$$L(\cdot) = -\sum_{i=1}^3 \frac{\partial}{\partial z_i} \left(b(z)k(z) \frac{\partial}{\partial z_i} \right) + \xi.$$

Розщеплення, що використовувались для розв'язання, записуються у вигляді таких операторів:

$$A_x = -\frac{\partial}{\partial z_1} \left(b(z)k(z) \frac{\partial}{\partial z_1} \right) + \frac{1}{2}\xi,$$

$$A_y = -\frac{\partial}{\partial z_2} \left(b(z)k(z) \frac{\partial}{\partial z_2} \right) + \frac{1}{2}\xi,$$

$$A_z = -\frac{\partial}{\partial z_3} \left(b(z)k(z) \frac{\partial}{\partial z_3} \right), \quad (14)$$

$$\varphi_1 = \varphi_2 = \varphi_5 = \varphi_6 = \frac{\varphi}{4},$$

$$\varphi_3 = \varphi_4 = 0.$$

Кожне з рівнянь (13) розв'язується методом скінченних різниць з використанням явно/неявної схеми Кранка–Ніколсона. Таким чином, можна істотно зменшити кількість обчислень – тепер вона залежатиме лінійно від розміру сітки. Рівняння, що записані для оператора A_z (14), можна без втрати точності звести до одного рівняння.

Паралельний алгоритм розв'язання задачі

Запропонуємо паралельний алгоритм розв'язання задачі на сітці $T \times N_x \times N_y \times N_z$ (де $N_x \times N_y \times N_z$ – внутрішні точки) на p процесорах.

1. Покласти $\xi \leftarrow \xi^0$.

2. Розв'язати задачу знаходження оптимального керування.

1°. Покласти $f \leftarrow f^0$.

2°. Розв'язати пряму задачу (3) методом сумарної апроксимації з цілим кроком з розпаралелюванням.

У загальному вигляді схему розв'язання для кожного кроку можна подати так, як показано в таблиці та на рис. 1 (допоміжні коефіцієнти для методу прогонки зберігаються, що дає можливість дещо зменшити кількість обчислень (до $5n - 10$ замість $8n - 7$); вартість формування правої частини рівняння врахована також). На початку кожному процесору q “на-

лежить” частина області $\left\{ (i, j, k), i = \overline{1, N_x}, j = \overline{1, N_y}, k = q \frac{N_z}{p} + 1, (q+1) \frac{N_z}{p} \right\}$.

При використанні такої схеми потрібна додаткова пересилка даних. Зокрема, після розв'язання другого рівняння процесор s має такі дані: $v_{ijk}^{(2,n+1)}$, $i = \overline{1, N_x}, j = \overline{1, N_y}, k =$

Таблиця. Розбиття задачі та завдання процесорам

Рівняння	Завдання процесору q	Обчислювальна складність
$\left(1 + \frac{\Delta t}{4} \left(A_1 + \frac{\xi_{n+1}}{2}\right)\right) v^{(1,n+1)} =$ $= \left(1 - \frac{\Delta t}{4} \left(A_1 + \frac{\xi_n}{2}\right)\right) u^n + \frac{\Delta t}{8} (f^{n+1} + f^n)$	Розв'язати $\frac{N_z N_y}{p}$ тридіагональних систем рівнянь розміру N_x	$(16N_x - 14) \frac{N_y N_z}{p}$
$\left(1 + \frac{\Delta t}{4} \left(A_2 + \frac{\xi_{n+1}}{2}\right)\right) v^{(2,n+1)} =$ $= \left(1 - \frac{\Delta t}{4} \left(A_2 + \frac{\xi_n}{2}\right)\right) v^{(1,n+1)} + \frac{\Delta t}{8} (f^{n+1} + f^n)$	Розв'язати $\frac{N_z N_x}{p}$ тридіагональних систем рівнянь розміру N_y	$(16N_y - 14) \frac{N_x N_z}{p}$
$\left(1 + \frac{\Delta t}{2} A_3\right) v^{(3,n+1)} = \left(1 - \frac{\Delta t}{2} A_3\right) v^{(2,n+1)}$	Розв'язати $\frac{N_x N_y}{p}$ тридіагональних систем рівнянь розміру N_z	$(12N_x - 14) \frac{N_y N_z}{p}$
$\left(1 + \frac{\Delta t}{4} \left(A_2 + \frac{\xi_{n+1}}{2}\right)\right) v^{(4,n+1)} =$ $= \left(1 - \frac{\Delta t}{4} \left(A_2 + \frac{\xi_n}{2}\right)\right) v^{(3,n+1)} + \frac{\Delta t}{8} (f^{n+1} + f^n)$	Розв'язати $\frac{N_z N_x}{p}$ тридіагональних систем рівнянь розміру N_y	$(16N_y - 14) \frac{N_x N_z}{p}$
$\left(1 + \frac{\Delta t}{4} \left(A_1 + \frac{\xi_{n+1}}{2}\right)\right) u^{n+1} =$ $= \left(1 - \frac{\Delta t}{4} \left(A_1 + \frac{\xi_n}{2}\right)\right) v^{(4,n+1)} + \frac{\Delta t}{8} (f^{n+1} + f^n)$	Розв'язати $\frac{N_z N_y}{p}$ тридіагональних систем рівнянь розміру N_x	$(16N_x - 14) \frac{N_y N_z}{p}$

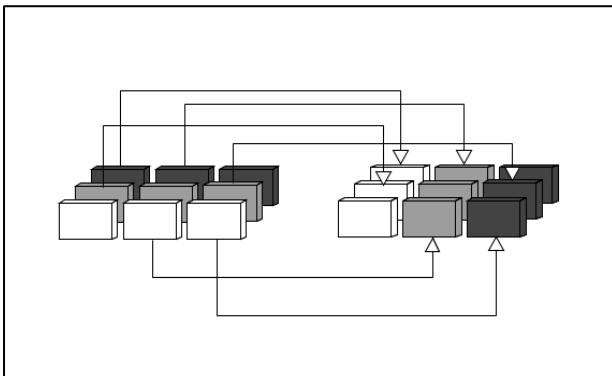


Рис. 1. Схема передачі даних між процесорами (області, що належать різним процесорам, позначені різними тонами; зліва – розбиття області за процесорами для другого рівняння, справа – для третього; стрілками показано, як передаються блоки даних)

$= s \frac{N_z}{p} + 1, (s+1) \frac{N_z}{p}$. Для розв'язання рівняння

(2) потрібні $v_{ijk}^{(2,n+1)}$, $i = s \frac{N_x}{p} + 1, (s+1) \frac{N_x}{p}$, $j =$

$= 1, \overline{N_y}$, $k = 1, \overline{N_z}$, тому для цього здійснюється глобальна пересилка даних – кожен процесор s надсилає кожному процесору q дані $v_{ijk}^{(2,n+1)}$, $i = s \frac{N_x}{p} + 1, (s+1) \frac{N_x}{p}$, $j = 1, \overline{N_y}$, $k = q \frac{N_z}{p} + 1, q \frac{N_z}{p}$ (див. рис. 1).

Зрозуміло, що аналогічна пересилка даних потрібна, коли здійснюється перехід від третього рівняння до четвертого.

Нехай $t(X)$ – час, потрібний для пересилки повідомлення розміру X між двома процесорами. Усі процесори можуть надсилати дані одночасно, тоді така пересилка даних для розв'язання всієї прямої задачі потребуватиме

$$C_{cd} = 2(T-1)(2p-3)t \left(\frac{N_x N_y N_z}{p^2} \right) \text{ операцій.}$$

Пояснимо цей вираз докладніше: за один обмін даними перший процесор обмінюється даними з кожним процесором за час

$t\left(\frac{N_x N_y N_z}{p^2}\right)(p-1)$; як тільки перший процесор обмінявся даними з i -м процесором, i -й процесор починає обмін даними з другим процесором, третім і т. д.; останній процесор чекає, доки перший не обміняється з ним даними. Потім останній процесор обмінюється даними з усіма іншими процесорами, що потребує $p-2$ кроків часу. Ця модель не є оптимальною, проте вона дає змогу уникнути проблем із взаємними блокуваннями.

Обчислювальна складність виражається формулою

$$T_d = (T-1) \left(78 \frac{N_x N_y N_z}{p} - 14 \left(2 \frac{N_y N_z}{p} + 2 \frac{N_x N_z}{p} + \frac{N_x N_y}{p} \right) \right).$$

На першому кроці, коли коефіцієнти невідомі, потрібно виконати

$$T_{1d} = (T-1) \left(105 \frac{N_x N_y N_z}{p} - \left(27 \frac{N_y N_z}{p} + 27 \frac{N_x N_z}{p} + 13 \frac{N_x N_y}{p} \right) \right).$$

Зауваження. Замість того, щоб пересилати дані на кроках 2 і 3, можна застосувати паралельний алгоритм прогонки (або алгоритм циклічної редукції [10]). Проте оскільки розв'язання кожного рівняння (13) призведе до розв'язання багатьох таких систем рівнянь методами, що передбачають міжпроцесорний обмін даними, то це може спричинити великі витрати на комунікацію, коли паралельна система характеризується значною латентністю.

3°. Перевірити умову

$$\|J_{li+1} - J_{li}\| / J_{li} > \varepsilon_1. \quad (15)$$

Якщо умова (15) виконується, перейти до п. 2,4°, інакше — до п. 3.

Для підрахунку інтегралів використовується традиційний метод паралельного підсумовування — на кожному процесорі підраховується часткова сума, а далі ці дані надсилаються головному процесору, який здійснює остаточний підрахунок. Потім цей процесор перевіряє виконання умови (15) та надсилає дані про те, чи потрібно завершувати виконання алгоритму

всім іншим процесорам. Складність такого алгоритму та витрати на комунікації записуються у вигляді формул

$$T_s = \frac{2N_x N_y N_z}{p} + p + 1, \quad (16)$$

$$C_s = t(1) + (p-1)t(1) = pt(1).$$

4°. Розв'язати спряжену задачу (11) (аналогічно кроку 1, враховуючи час на обчислення значення для останнього кроку) за час

$$T_{ca} = (T-1) \left(60 \frac{N_x N_y N_z}{p} - 14 \left(2 \frac{N_y N_z}{p} + 2 \frac{N_x N_z}{p} + \frac{N_x N_y}{p} \right) \right) + \frac{N_x N_y N_z}{p}.$$

Кількість операцій при цьому становитиме

$$C_{ca} = 2(T-1)(2p-3)t\left(\frac{N_x N_y N_z}{p^2}\right).$$

5°. Модифікувати параметр джерела згідно з

$$(10) \text{ за час } T_m(p, N_x, N_y, N_z, N) = 2N \frac{N_x N_y N_z}{p}.$$

Перейти до п. 2,2°.

3. Перевірити виконання умови для функціонала, що відповідає за параметр процесу (інтегрування здійснюється аналогічно). Цей функціонал залежить від градієнта, що апроксимується центральними різницями, потрібний додатковий обмін даними між процесорами (кожен процесор рангу $q \neq 0$ надсилає процесору $q-1$ блок даних $\{u_{ijk}^n, i = \overline{1, N_x}, j = \overline{1, N_y}, k =$

$$= q \frac{N_z}{p} + 1, n = \overline{1, T}\}, \text{ і кожен процесор } q \neq p-1$$

надсилає процесору $q+1$ блок даних $\left\{ u_{ijk}^n, i = \overline{1, N_x}, j = \overline{1, N_y}, k = (q+1) \frac{N_z}{p}, n = \overline{1, T} \right\}$. Вва-

жаючи, що кожна пересилка даних виконується всіма процесорами одночасно, отримуємо оцінку

$$C_{add} = 2t(N_x N_y).$$

Для обчислення функціонала потрібен час (комунікації — аналогічно (16))

$$T_{ms} = 18T \frac{N_x N_y N_z}{p} + p + 1.$$

Якщо умова не виконується, розв'язок знайдено. Якщо виконується, перейти до наступного пункту.

4. Розв'язати спряжене рівняння (9).

Розв'язання потребує (із врахуванням часу на формування правої частини та враховуючи операцію порівняння) часу

$$T_{ma} = (81T - 78) \frac{N_x N_y N_z}{p} - 14(T - 1) \left(2 \frac{N_y N_z}{p} + 2 \frac{N_x N_z}{p} + \frac{N_x N_y}{p} \right)$$

і кількість операцій

$$C_{adj} = 2(2p - 3)(T - 1) \left(\frac{N_x N_y N_z}{p^2} \right).$$

5. Модифікувати параметр перешкоди згідно з (8). Це потребуватиме часу

$$T_{ml} = (7T - 2) \frac{N_x N_y N_z}{p}.$$

6. Перейти до п. 2. Час виконання алгоритму як функції від кількості процесорів можна подати в загальному вигляді як $f(p) = \frac{\alpha}{p} -$

$-\frac{\beta}{p^2} + \gamma p + \delta$, де $\alpha, \beta, \gamma, \delta > 0$ — функції розміру сітки дискретизації та кількості ітерацій (тут для оцінки часу передачі повідомлень застосовувалася модель Хокні [11]).

Обчислювальні експерименти

Обчислювальний експеримент виконано із застосуванням кластера ЕОМ НТУУ “КПІ” з використанням MPI. Було отримано результати, що свідчать про прискорення обчислень при застосуванні паралельного алгоритму (рис. 2). Результати демонструють, що прискорення

($S = \frac{T_i}{T_p}$, де T_i — час виконання алгоритму на

i процесорах) алгоритму погіршується із збільшенням кількості процесорів, бо це призведе до значних комунікаційних витрат [12].

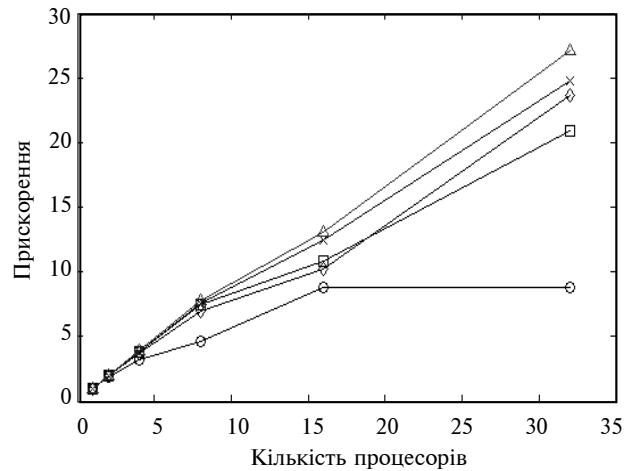


Рис. 2. Прискорення алгоритму на сітках різного розміру: —○— 32×32×32; —◇— 64×64×64; —□— 96×96×96; —×— 128×128×128; —△— 160×160×160

Цікаво, що для сітки розміру 32×32×32 прискорення на 32 процесорах виявляється майже таким, як і на 16 процесорах, що пов'язано з великою витратою часу на комунікації.

Висновки

Запропонований у статті розпаралелений алгоритм керування одностороннім процесом може бути використаний в умовах, коли моделювання та керування процесом виконуються в реальному часі і висувають підвищені вимоги щодо швидкості обчислень або при виконанні досліджень поведінки даних процесів із використанням великих розмірностей сіток. Отримані результати показують, що із збільшенням кількості процесорів прискорення алгоритму зменшується, що свідчить про те, що досить багато часу йде на комунікації.

Як розвиток досліджень у цій галузі було б доцільним запропонувати паралельні варіанти алгоритмів моделювання та керування односторонніми процесами з адаптивним ущільненням сітки дискретизації, що дало б можливість ще більше підвищити ефективність застосування паралельного підходу.

М.С. Качановская, И.В. Жданова

РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЙ В ЗАДАЧЕ УПРАВЛЕНИЯ ОДНОСТОРОННИМИ ПРОЦЕССАМИ ДИФФУЗИИ И ТЕПЛОМАССОБМЕНА

Разработан параллельный алгоритм для моделирования и управления односторонними процессами диффузии и тепломассообмена. Алгоритм экспериментально апробирован для задачи управления односторонним процессом фильтрации нефти с граничным градиентом давления. Модель процесса рассматривается в виде вариационного неравенства, для решения которого использован подход функциональной параметризации. Параллельные вычисления выполнены с использованием кластера НТУУ "КПИ".

M.S. Kachanovska, I.V. Zhdanova

COMPUTING PARALLELIZATION IN CONTROL OF UNILATERAL PROCESSES OF DIFFUSION AND HEAT-MASS EXCHANGE

In this paper, we develop the parallel algorithm for modeling and control of unilateral processes of diffusion and heat-mass exchange. The algorithm is experimentally tested for control of unilateral process of oil filtration with limited pressure gradient. The process model is considered as a variation inequality. We use functional parameterization for its solution. Using the cluster of Kyiv Polytechnic Institute, we do parallel calculations.

1. *Згуровский М.З., Мельник В.С., Новиков А.Н.* Прикладные методы анализа и управления нелинейными процессами и полями. — К.: Наук. думка, 2004. — 588 с.
2. *Дюво Г., Лионс Ж.-Л.* Неравенства в механике и физике. — М.: Наука, 1980. — 384 с.
3. *Згуровский М.З., Новиков А.Н.* Анализ и управление односторонними физическими процессами. — К.: Наук. думка, 1996. — 328 с.
4. *Жданова І.В., Новіков О.М.* Модифікована схема розв'язування скінченноелементних моделей одного класу методом перерізів // Наукові вісті НТУУ "КПІ". — 2003. — № 5. — С. 138–147.
5. *Лионс Л.* Оптимальное управление системами, описываемыми уравнениями с частными производными. — М.: Мир, 1972. — 414 с.
6. *Макконелл Дж.* Основы современных алгоритмов. — М.: Техносфера, 2004. — 368 с.
7. *Марчук Г.И.* Методы расщепления. — М.: Наука, 1988. — 536 с.
8. *Параллельные вычисления* / Под ред. Г. Родрига; Пер. с англ. — М.: Наука, Гл. ред. физ.-мат. лит.-ры, 1986. — 376 с.
9. *Самарский А.А., Вабищев П.Н.* Вычислительная теплотопередача. — М.: Эдиториал УРСС, 2003. — 784 с.
10. *Allmann S., Rauber T., Runger G.* Parallel and Distributed Processing // Proceedings. Ninth Euromicro Workshop on Volume, Issue, 2001 Page(s). — 290 p.
11. *Hockney R.* The communication challenge for MPP: Intel Paragon and Meiko CS-2. Parallel Computing, 20(3): 389–398, March 1994.
12. *Ahmar Abbas.* Grid Computing: A Practical Guide to Technology and Applications. — Charles River Media, 2004.

Рекомендована Радою
Фізико-технічного інституту
НТУУ "КПІ"

Надійшла до редакції
27 травня 2009 року